Towards a Greener Web

Elia Lejzerowicz eel2157

December 16, 2022

1 Introduction

With the rise of online services such as e-commerce, streaming video, and social networking platforms, the use of the internet has expanded significantly in recent years. These services rely on large IT infrastructures, such as data centers and servers, which consume and emit a considerable amount of energy and carbon dioxide. In fact, the Internet is responsible for 2 percent of global carbon emissions, which is equivalent to the amount produced by the aviation industry. As this number continues to rise alongside the internet's rapid expansion, there is an urgent need to find methods to stop and mitigate its effect. This project intends to assist websites in reducing their carbon footprint through the application of targeted recommendations and guidelines. As the configuration of a website has a direct impact on its carbon footprint, we analyze the design of hundreds of websites, evaluating their Eco-Impact and identifying the primary factors that could be modified to reach a higher Eco-score. This study is an expansion to the EcoIndex project (GreenIt.fr), and an active contribution the EcoSonar (ecosonar.org) open-source project at Accenture Technology. The purpose is to raise awareness of environmental concerns and eco-design techniques, quantify the carbon footprint of digital services, and obtain a solution for environmental and performance monitoring.

2 Literature review

Concerns were raised over the environmental impact of the internet and digital technology. In recent years, there has been a rise in the demand for electricity due to the skyrocketing use of internet services, which has led to an increase in harmful CO2 emissions. There have been publications with provocative titles such as "Your website is destroying the world" and studies on the issue. Consequently, the environmental effect of our growing dependence on digital technologies is becoming more difficult to ignore.

The environmental footprint of a webpage is composed of its water usage and greenhouse gas emissions. A web page's environmental footprint is the effect that displaying the page has on the environment. This impact is a result of two factors: the greenhouse gases emitted during the production and use of the equipment required to view the page, and the water consumption associated with that equipment. In fact, a large amount of water is needed for the extraction and purification of the metals required to make the equipment. The emissions of an average page can be used to calculate these values, which can then be changed based on the specifics of the page being looked at. By understanding the environmental footprint of a web page, we can make informed decisions about how to reduce its impact on the environment.

Various methods have been used to estimate the carbon footprint of the internet, or the amount of carbon dioxide emissions caused by internet usage. However, there is no exact method for calculating the Internet's carbon footprint, and estimates vary considerably. In this study, we chose to rely on a formula provided by the EcoIndex project. This project provides users with the EcoIndex tool to evaluate the environmental impact and sustainability of a website. Users would

enter the URL of the webpage they wish to analyze, and the tool would calculate an EcoIndex grade to determine the page's carbon footprint. This research considers that the carbon impact of a website should be calculated based on three different metrics: The size of the domain, the number of requests and the size of the response, which should represent the carbon impact. In addition, the tool provides an audit of the webpage by analyzing multiple metrics relative to the conception of the site, and identifying which recommendations were not implemented (Table 1). However, it is difficult for a website to decide which metric to prioritize in order to achieve a higher grade with the least amount of changes. As a result, this research aims to inform websites about which recommendations they should prioritize. Consequently, the objective of this study is to expand upon the previously published EcoIndex.fr project and to actively contribute to the EcoSonar project at Accenture, which aims to offer websites a recommendation system that could be based on this analysis.

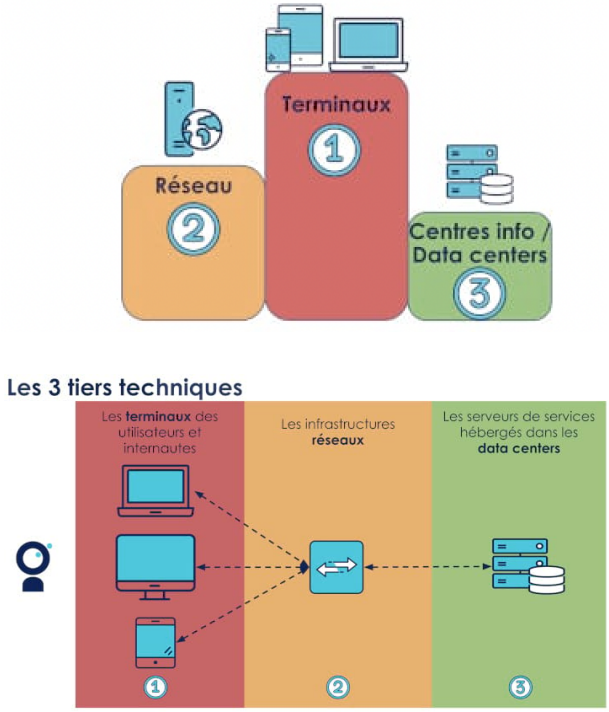


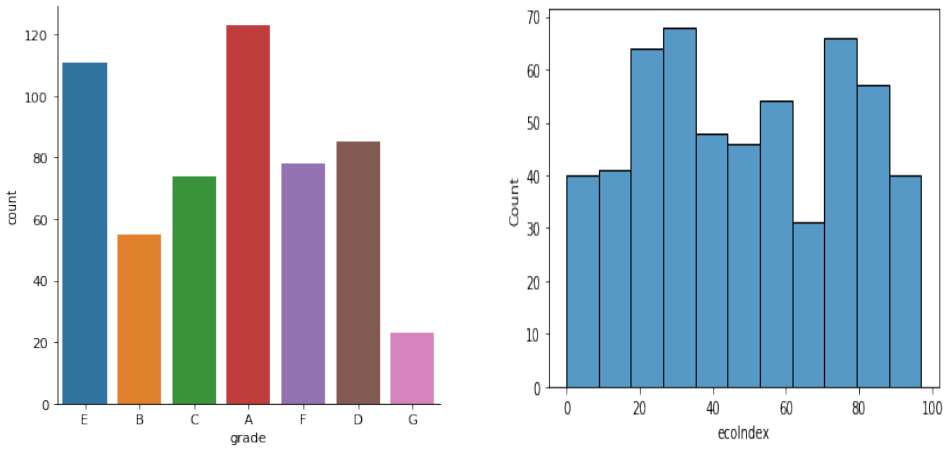
Figure 1: IT infrastructures breakdown responsible for the environmental impact

3 Data Processing and Analysis

A dataset was gathered with the assistance of Accenture Technology France for their ongoing Ecosonar project. Using two web auditing tools (Google Lighthouse and GreenIt, both Chrome extensions), the team analyzed 500 distinct websites and collected their metrics, which are configuration-related data. Each website is assigned an eco index score (ranging from 0 to 100) and a letter grade (ranging from A-G) that represents its carbon footprint (the highest grade representing the least amount of carbon emitted).The dataset consists of approximately 50 types of recommendations, where half are retrieved from the GreenIt analysis and the remaining half from Google Lighthouse. The greenIt audit tool provides websites the indication on whether the best practices are well or well not implemented. When gathered, these informations needed to be preprocessed since the original values were measured differently for each practice. As a result, the team at Accenture focuses on implementing a system that assigns a score between 0 and 100 based on how well a website implements the best practices. The Google Lighthouse analysis is

also included into the dataset, and provides information on the speed and performance along with additional best practices ranging from 0 to 100, indicating the level of implementation of the recommendation. As a result, the dataset consists of the best practices metrics from the GreenIt and Google lighthouse auditing tools. We aim to search for the link between the EcoIndex score and the implementation of the best practices in order to rank them from most significant to least significant. This enables us to provide websites with the most effective recommendations.

After the data was pre-processed, we analyzed it. We noticed that the dataset is very well balanced as each grade is represented. We also calculated the correlation between the feature which is included in the Github. Note that the HttpRequest recommendation feature was deliberately removed from the analysis as it is directly correlated to the number request and therefore consistently reached the highest feature importance for all models, resulting in a higher and misleading accuracy.



(a) Eco Grade **(b)** Eco-Index

Figure 2: Website Carbon Impact

The Eco-Index score is calculated according to the following formula:

$$100 - \frac{3 * Qtil(DOM) + 2 * Qtil(HTTP) + 1 * Qtil(KO)}{6} \quad (1)$$

The formula accepts the number of DOM elements, the number of HTTP requests, and the page's weight as inputs. "DomSize" represents the set of the page's display structure (successive HTML or XML tags). The greater the complexity of the Document Object Model (DOM), the more potent the terminal must be, and the more RAM or CPU power it requires. The "ResponseSize" (KO) indicates the size of a page downloaded from the server to the terminal. "NB request" (Http) represents the total number of HTTP requests between the servers and the terminal. In addition, the formula assigns a specific weight to each component. DomSize has the highest value, followed by HttpRequest and responseSize. Various micro and macro analyses highlight the preponderance of the "internet user" third party in terms of environmental impacts, particularly during equipment production. Due to this, it is comparatively heavier than the others. The grade associated with the EcoIndex is a representation of its carbon impact. It is ranked on a scale from A to G (European standard). The higher the grade, the better the eco-conception of the website. Consequently, we would like to give advice to websites on how to reach an A grade. As a result, we are searching for the best practices that would have the most influence on the score, in order to sort the recommendations from the most to the least impactful.

4 Methods and Application

In order to uncover the feature importance corresponding to the essential practices one should implement to reach the highest EcoIndex score, four machine learning regression tasks are conducted, and two different models are used and evaluated for each of those: Random Forest and XGboost. The first model is a regression task that predicts the EcoIndex score and computes the most significant feature. Since we know that the EcoIndex is derived from the Number of Request, the DomSize, and the ResponseSize, we perform three other regressions tasks to determine the most important features of these three metrics. This is crucial, as the ultimate aim is to develop a recommendation algorithm that can provide customized recommendations for websites based on the value of each of those three metrics. Therefore, our goal is to identify the primary contributors to that amount, which corresponds to the feature importance of the models.

The selection of the machine learning methods was deliberate in order to achieve our research objectives. Since our primary objective is to identify the feature importance, a decision tree is an excellent choice for presenting this data. Moreover, random forest can accommodate multi-colinearity. This is because a random forest model is an ensemble model, consisting of multiple decision trees trained on a random subset of the data. Since each decision tree is trained on a distinct subset of the data, the trees are able to learn from the correlated features and determine which ones are most crucial for making predictions. This enables the random forest model to capture the importance of each feature accurately, even when the features are highly correlated, which in our case is highly important considering the several correlated values calculated in the data Analysis.

Even though these two techniques are similar in that they both use ensembles of decision trees to make predictions, they differ in the manner in which they grow their trees and split its nodes, which can affect their performance and usability. In fact, Random forests employ a technique known as bagging, in which each tree in the ensemble is trained using a random subsample of the training data. In contrast, XGBoost employs a technique known as gradient boosting, in which each tree in the ensemble is trained to correct the errors made by the previous tree.

The cross validation technique was also used as a mean to evaluate the performance of the models on unseen data and to ensure the robustness of our models. This was done by splitting several times the original dataset into 5 subsets, training the model on one subset and evaluating its performance on the other. Additionally, it helps us tune the hyper parameters of the model, such as the regularization parameter, to get the best possible performance.

When using the random forest model, hyper-parameter tuning was performed using grid search. The following hyper-parameters were tuned: The number of trees in the forest and the maximum depth of each tree. The number of trees controls the number of decision trees that are included in the random forest model. Increasing the number of trees can improve the model's ability to capture complex patterns in the data, which can lead to improved accuracy. However, too many trees can make the model slow and unwieldy, so it is important to strike a balance between accuracy and computational efficiency. The maximum depth of each tree controls the maximum number of levels that each decision tree in the random forest can have. Increasing the maximum depth of each tree will allow the model to make more complex decisions, which can potentially improve the accuracy of the model. However, too much depth can lead to over fitting, so it is important to carefully tune this hyper parameter to find the right balance.

When using the XGboost model, the following hyper-parameters were tuned: the learning rate, the number of estimators and the minimum number of splits. The learning rate hyper parameter controls the step-size in updating the weight of each tree at each iteration. It is a regularization parameter that controls the magnitude of the updates to the weights. The minimum number of samples split controls the minimum number of samples required to split a decision tree node. The number of estimators determines the number of trees within the model.

In light to the topic that was covered in class, the Bayesian optimization method was also utilized. This method is especially useful when it is difficult to measure the function being optimized, such as when simulating or testing. It employs a probabilistic model that strikes a balance between exploration and exploitation, balancing the need to search the search space for good answers with the need to use what it has learned so far to improve the answers it has already found. One of the advantages of Bayesian optimization is its ability to circumvent search space limitations. To accomplish this, a probabilistic model is employed to predict how the function will behave, and this model is updated as new data is gathered. This hyper parameter approach is only applied on the EcoIndex prediction since we aim to strengthen the predictive aspect of this particular model.

Recursive Feature Elimination (RFE) is a common machine learning feature selection approach. It recursively removes the least significant features from the model until the required number of features is obtained. This method could offer several advantages, including a decrease in the model's complexity, enhancing its generalizability, and enhancing its computing efficiency. RFE can be effective in a variety of circumstances and in our case, for our high-dimensional data. RFE may enhance the performance and interpretability of machine learning models by eliminating unnecessary or duplicate information.

5 Feature Importance

The significance of each model's features has been assessed. The feature importance quantifies the contribution of each feature to a machine learning model's predictions. This information is crucial to our analysis as we especially aim to uncover what features participated the most to our predictions. The feature importance was calculated using two different methods. First, the feature importance directly provided from the random forest and the XGboost model.

Random forest:

$$importance(n) = w_n C_n - \sum_{n' \text{child of } n} w_{n'} C_{n'}$$

where

• wn: is the fraction of samples that reach node n

• Cn: is the variance of the target of samples that reach node n

A feature's importance in a Random Forest model can be calculated as the sum of its' importances in all the tree's it's sampled in.

Shapley Additive Explanations:

The second method used to compute the feature importance is SHAP (Shapley Additive Explanations).

Shapley value:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

where g is the explanation model, $z' \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size and $\phi_j \in R$ is the feature attribution for a feature j, the Shapley values. To calculate Shapley values, we simulate that only certain feature values ("present") are active and others ("absent"). The depiction as a linear model of coalitions is a technique for calculating (phi). For

x, the instance of interest, the coalition vector x' is a vector of all 1's, i.e. all feature values are "present". The formula is as follows:

SHAP:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j$$

SHAP is a method for calculating the significance of features that is based on the Shapley values from game theory. This method can provide a more precise and thorough assessment of feature importance than other methods, such as permutation importance and tree-based model feature importances. These values take into account how much each feature adds to the overall accuracy of the model as well as how the features affect each other and was therefore chosen as our final feature selection.

Nonetheless, it is advantageous to use the two distinct feature importance computation methods in order to compare their differences as well as those between the two tree-based models. If the importance of the best features corresponds to each of these, it will be easier for us to identify the most important features.

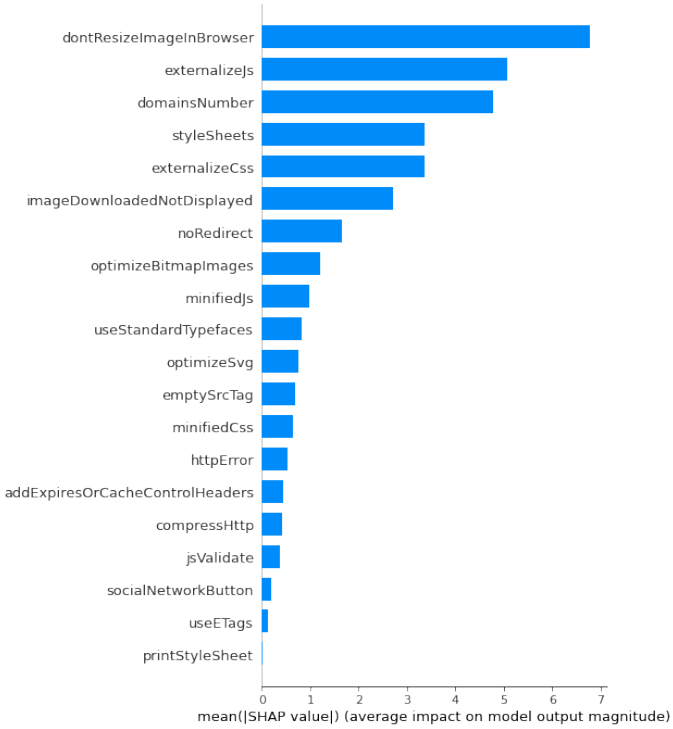
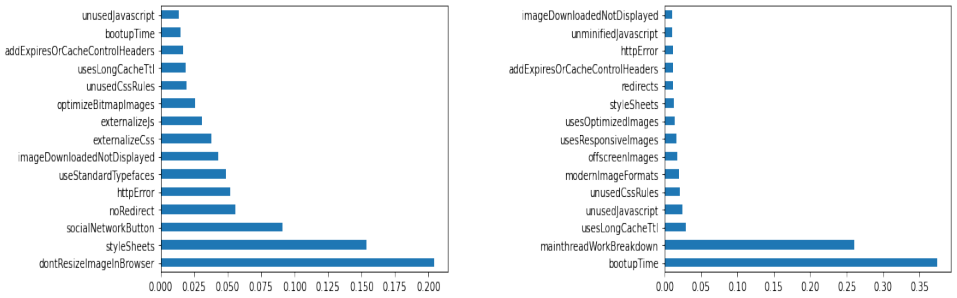


Figure 3: EcoIndex Feature Importance

The figure 4 are the feature's importance obtained through the prediction of the EcoIndex. The figure 5 are the feature's importance obtained trough the prediction of the Number of Request and the DomSize. The Response size was omitted on purpose since the accuracy of the model is too less relevant.

6 Recommendations

After calculating the most significant features responsible for the most relevant recommendations, we aim to construct a recommendation algorithm. This algorithm would determine the best



(a) Number of Request **(b)** Domsize

Figure 4: Metric's Feature Importance

recommendations to implement for a given website, in descending order of impact and complexity. In fact, it accounts for the feature importance score, its weight according to the EcoIndex formula, and the implementation difficulty (checked with Accentur). A method has been developed for displaying these recommendations and estimating the potential score increase that may be attained. The method receives the URL of a website as input and returns an ordered list of recommendations as output. The algorithm looks at each row and analyses the value of the 3 different metrics. If the metric is under the target, it will provides the recommendation retrieved from the feature importance of that specific target.

	mediane	Target
DomSize	2.41	1.024
ResponseSize	693	600
Number of Request	78	40

Figure 5: Metrics

7 Results and discussion:

Figure 7 summarizes the results obtained from the different machine learning models. It contains the performance of each model for the four regression tasks as well as the feature importance retrieved from SHAP. From the findings represented in the table, we observe that the predictive EcoIndex model provides the best results in comparison to the other models and is sufficient to capture the most important features. Unfortunately, the results obtained from the three other predictive models, the number of request, the domSize and the responseSize are disappointing. Each of these models overfits and provide poor results on the testing data. Subsequently, we debated on whether the initial algorithm we envisioned should rely on the best practices retrieved from their feature importance. As a result, the recommendation algorithm has therefore been implemented not only taking into account the feature importance from those three models but also and mostly from the ones retrieved from the EcoIndex model. Regarding the different models, we observe that the XGboost performs better across the models and we witness that the Bayesian Optimization has improved the score of the EcoIndex model. On the other side, RFE didn't provide any improvement and thus was not included in the other models.

To evaluate the overall performance of our models and determine whether the recommended features have a significant impact on the EcoIndex score, we simulated the effect of applying the targeted recommendations to the websites with a low EcoIndex score. Consequently, a new artificial dataset was generated by increasing the score of relevant recommendations based on the importance of the feature. Only the top three recommendations obtained with the EcoIndex

	Random Forest +GridSearch +cv	XGboost +GridSearch	Best Feature Importance
EcoIndex	Training: 0.90 Testing: 0.71	Training: 0.82 Testing: 0.76 **Bayesian OPT:** Testing: 0.78	"dontResizeImageInBrowser" "ExternalizeCSS/JS" "DomainsNumber"
NbRequest	Training: 0.93 Testing: 0.55 **RFE:** Training: 0.94 Testing: 0.52	Training: 0.82 Testing: 0.53	"dontResizeImageInBrowser" "domainsNumber" "styleSheets"
DomSize	Training: 0.93 Testing: 0.56	Training: 68 Testing: 54	"MainThreadWorkBreakdown" "BootupTime"
ResponseSize	Training: poor Testing: poor	Training: poor Testing: poor	Not relevant

Figure 6: Models peformance

model were changed for this experience. We also use this best predictive model (XGboost model optimized with bayesian optimization) to predict the EcoIndex based on both the actual and the modified dataset. Consequently, we are able to compare the scores, and thankfully, observe a clear improvement in the EcoIndex score for each website with a poor grade. The figure 8 showcases the predictions based on our artificial dataset as opposed to the previous dataset. This clearly validates our hypothesis and confirms that only few recommendations could be applied in order to substantially increase the score.

8 Conclusion and Future Work:

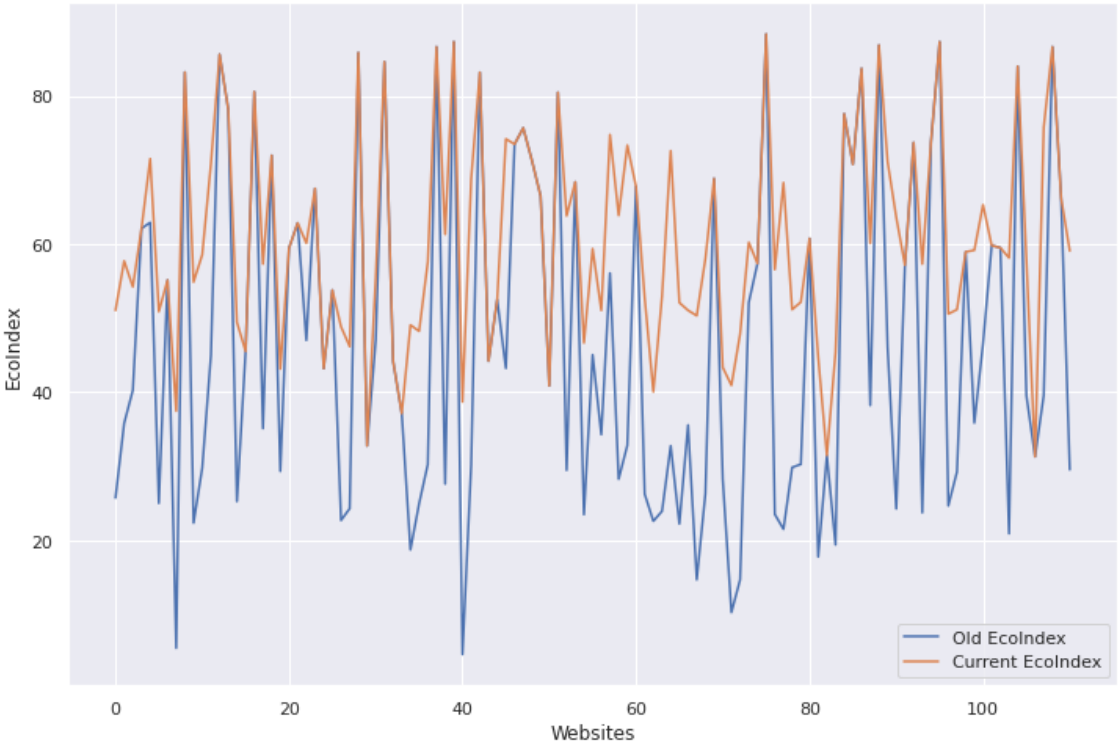
The purpose of this research was to simplify the eco-design configuration of websites in an effort to reduce their environmental impact. We collected data from 500 distinct websites and analyzed their eco-concept and level of implementation of best practices. Multiple machine learning analyses were conducted to estimate their carbon footprint reflected in their EcoIndex score. We compiled the most important features representing the most impactful recommendations to reduce the carbon impact. We performed an experiment with a synthetic dataset and predicted the outcome of applying the fewest number of recommendations with the greatest impact on the score. We concluded that it would be possible to achieve a higher score if only these target practices were modified. We hope that this study has contributed to a greater awareness of a simple and effective eco-concept for websites and will aid developers in this endeavor. In addition, we hope that the Ecosonar team at Accenture will find this research useful and incorporate the findings into the development of their application that aims to provide targeted recommendations to websites.

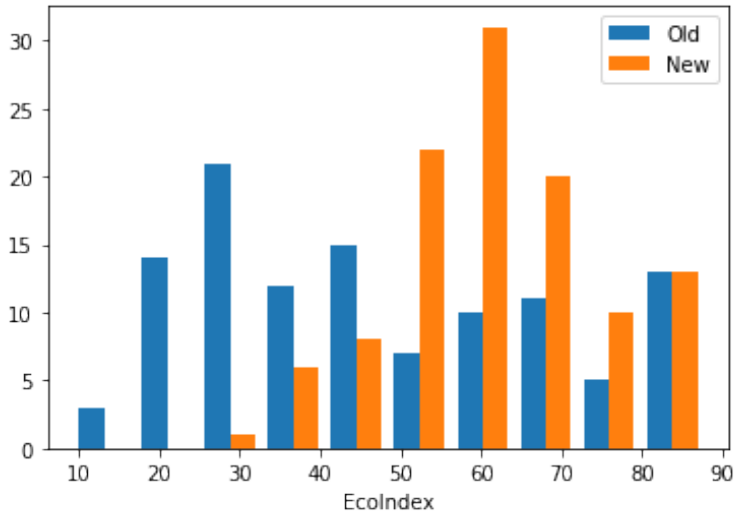
9 Appendix:

The regression model of the EcoIndex outputed multiple important features. We will analyze and expand on the most important ones according to our models.

Nb Request: Reducing the number of page requests can make a website run better and give users a better experience. By reducing the number of HTTP requests that the browser needs to make, the page will load faster, providing a better user experience. Techniques such as merging static files, using CSS sprites, and using glyphs instead of images can help reduce the number of page requests. Additionally, maximizing the browser's cache can reduce the number of requests that

Figure 7: Impact of targeted recommendation accross websites





need to be made to the server. These efforts can improve the performance and user experience of a website, as well as reduce its environmental impact.

Domains number: Hosting parts of a web page on a single domain can help a website run better. When a browser has to connect to multiple domains to load a page, it can slow down the time it takes to show the page. By putting all of a page's resources on one domain, the browser only has to make one connection, which can make the page load faster.

Don't resize image in browser: Resizing images using HTML height and width attributes can waste bandwidth and CPU power. When an image is resized using these attributes, the original size image is still sent, even if it is displayed at a smaller size. This can result in unnecessary data being transferred, increasing the page's download time and using more bandwidth. It's best to resize images using image editing software, such as Photoshop, before uploading them to the website. This makes sure the image is the right size before sending it to the browser. Also, preventing users from adding images with a WYSIWYG editor can help cut down on the amount of unnecessary data that is transferred. Overall, a website can work better and be more efficient if images are resized correctly and the number of images that can be added is limited.

Externalizing CSS/JS: On the development side, moving CSS and JavaScript files outside of the website can make it run faster. By separating CSS and JavaScript code from the HTML code of the page, the browser can store these files locally and not have to ask for them again on other pages. This can reduce the amount of data that needs to be sent, improving the performance of the website. Also, moving these files outside of the website can make it easier to maintain and update the code, since changes can be made in one place instead of on each page. Overall, moving CSS and JavaScript files outside of a website can make it run faster, be easier to maintain, and be more efficient.

Minimize main-thread work: In a web browser's renderer process, the main thread is in charge of running the code needed to show a web page and respond to user actions. It is responsible for parsing HTML and CSS, building the DOM, and executing JavaScript code. It is important to make sure that the main thread doesn't have too much work on it, as this can slow things down and make the user experience bad. To avoid this, it is recommended to offload long-running tasks or computations to worker threads or other background processes, and to use techniques such as code splitting and lazy loading to reduce the amount of code that needs to be executed on the main thread.

References

[GreenIT] reenIT.fr - la communauté des acteurs de la sobriété numérique et du numérique responsable (green it, low-tech numérique, écoconception web et de service numérique, etc..) Green IT. Available at: https://www.greenit.fr/.

[Ecosonar] aupais, A. Accenture Ecosonar, EcoSonar. Available at: https://ecosonar.org//

[WebFX] mily Carternbsp; Powering the internet: Your virtual carbon footprint [infographic], WebFX. Available at: https://www.webfx.com/blog/marketing/carbon-footprint-internet/

[OCTO] REPY, P. (2022) Sous le capot de la mesure ecoindex !, OCTO Talks ! Available at: https://blog.octo.com/sous-le-capot-de-la-mesure-ecoindex/

[RESET] ur Digital Carbon Footprint: What's the Environmental Impact of the Online World? (2022) RESET.ORG. Available at: https://en.reset.org/our-digital-carbon-footprint-environmental-impact-living-life-online-12272019